

### create a new repository on the command line

echo "# jkljkl" >> README.md

git init 🡨-empieza a trackear el directorio actual en git

git add README.md

git commit -m "first commit" 🡨 es un mensaje que se deja enntre cada actualizacion

git branch -M main <- se crea una nueva rama, Main corresponde a la principal

git remote add origin <https://github.com/DaniloPaz01/jkljkl.git> <- se agrega el origen remote de la nube de github (url + Nick / nombre del repo)

git push -u origin main <- sube el repo actual a la github

**or push an existing repository from the command line**

git remote add origin https://github.com/DaniloPaz01/jkljkl.git

git branch -M main

git push -u origin main

**Version**

git –version <- muestra la version de git

**Configuracion global**

git config -–global user.name “Danilo” <- setea el usuario global para toda la maquina actual donde este instalado git

git config –global user.email [danilopaz@gmail.com](mailto:danilopaz@gmail.com) <- setea el mail global para la pc actual

git config –list <- muestra todas las configuraciones de git por consola

**Ayuda**

Git help<verb> , git help config

Git config –-help

**Inicializar un repo con código existente**

Git init

Git status <- muestra el estado de los elementos dentro del repo (trackeado para comitear,cambios etc)

.gitignore <- es un archivo donde se indica cuales elementos del repositorio se van a ignorar para no cargarlos a la nube se pueden usar comodines con las extensiones o simplemente nombrarlos

Git add archivo1 <- prepara el elemento para el comit

Git add –A <- prepara todos los elementos del repositorio para el comit

Git commit –m “un mensaje” <-sube los archivos a la nube con un mensaje

Git status <- muestra el estado del repo

Git log <- muestra los logs dfe la actividad reciente

Git diff <- muestra los cambios hechos entre los archivos anteriores y actuales

**Clonando un repo**

Git clone <url> <dirección de donde lo queremos clonar>

Git clone ../remote/repo\_git . <- notar que el punto clona en el lugar actual

**Ver la información del repositorio remoto**

Git remote –v <-muestra info del repo

Git branch –a <-muestra info de la ramificacion actual

**Subiendo los cambios al repositorio**

Git pull <origen> <nombre de la rama>

Git pull origin main 🡨 descarga los cambios que se pudieron haber hecho en el código

Git push origin main 🡨 sube los cambios hechos por mi al repo

**Creando una rama para una nueva característica**

Git branch <nobmre>

Git branch logicaDeGuardado

**Cambiando de rama**

Git checkout logicaDeGuardado

Git branch 🡨 muestra la rama actual

Git push –u origin logicaDeGuardado

Git branch –a 🡨 muestra todas las ramificaciones del repo

**Mergeando ramificaciones**

Git checkout main <- cambiar al main

Git pull origin main

Git branch –merge <- muestra las ramas que se mezclaron hasta ahora

Git merge lógicaDeGuardado <- mergea con la rama main

Git push origin master

**borrando ramificaciones**

Git branch -d logicaDeGuardado

Git branch –a

Git push origin –delete logicaDeGuardado